Improving regularized singular value decomposition for collaborative filtering

Arkadiusz Paterek <paterek@mimuw.edu.pl>
Institute of Informatics, Warsaw University

August 12, 2007
7.04% improvement over Netflix’s recommendation system
The goal

- to predict users’ preferences for movies, as accurately as possible

Outline of the approach

- take many well predicting methods
- combine their results with linear regression on the test set
The most important methods in the ensemble

- regularized SVD - by Simon Funk
- *New*: regularized SVD with biases
- clustering users using K-means
- postprocessing SVD results with item-item K-nearest neighbors
- *New*: postprocessing SVD with kernel ridge regression
- *New*: weighted linear model for each item
- *New*: methods similar to SVD, but with fewer parameters

Together they give 6.34% improvement.
The 7.04% solution is a result of linear regression of 56 predictors and 63 two-way interactions between them, plus partial cross-validation.
Regularized SVD

Predictions for user $i$ and movie $j$:

$$\hat{y}_{ij} = u_i^T v_j$$

where $u_i$ and $v_j$ are $K$-dimensional vectors of parameters.

Parameters are estimated by minimizing the sum of squared residuals, one feature at a time, using gradient descent with regularization and early stopping:

$$r_{ij} = y_{ij} - \hat{y}_{ij}$$

$$u_{ik} += \text{lrate} \times (r_{ij} v_{jk} - \lambda u_{ik})$$

$$v_{jk} += \text{lrate} \times (r_{ij} u_{ik} - \lambda v_{jk})$$
Regularized SVD with biases

We add biases to the regularized SVD model, one parameter $c_i$ for each user and one $d_j$ for each movie:

$$\hat{y}_{ij} = c_i + d_j + u_i^T v_j$$

Weights $c_i$, $d_j$ are trained simultaneously with $u_{ik}$ and $v_{jk}$:

$$r_{ij} = y_{ij} - \hat{y}_{ij}$$

$$u_{ik} += lr* (r_{ij} v_{jk} - \lambda u_{ik})$$

$$v_{jk} += lr* (r_{ij} u_{ik} - \lambda v_{jk})$$

$$c_i += lr* (r_{ij} - \lambda_2 (c_i + d_j - \mu))$$

$$d_j += lr* (r_{ij} - \lambda_2 (c_i + d_j - \mu))$$
Postprocessing SVD with kernel ridge regression

- idea: discard all weights $u_{ik}$ after training and, for each user $i$, predict $y_{ij}$ using $x_{j2} = \frac{v_j}{||v_j||}$ as predictors
- using ridge regression gives similar results to the reg. SVD
- better prediction can be obtained by using kernel ridge regression with Gaussian kernel
- predictions in kernel ridge regression:
  \[ \hat{y}_j = K(x_j^T, X)(K(X, X) + \lambda I)^{-1} y \]
- in place of the scalar product we use a Gaussian kernel:
  \[ K(x_j^T, x_k^T) = \exp(2(x_j^T x_k - 1)) \]
Weighted linear model for each item

Separate weighted linear model for each movie \( j \):

\[
\hat{y}_{ij} = m_j + e_i \times \sum_{j_2 \in J_i} w_{j_2}
\]

SVD-based methods with fewer parameters

Two models with \( O(MK) \) parameters (\( M \) movies, \( K \) features):

\[
\hat{y}_{ij} = c_i + d_j + e_i \sum_{k=1}^{K} v_{jk} \sum_{j_2 \in J_i} w_{j_2k}
\]

\[
\hat{y}_{ij} = c_i + d_j + \sum_{k=1}^{K} v_{jk} \sum_{j_2 \in J_i} v_{j_2k}
\]
<table>
<thead>
<tr>
<th>Predictor</th>
<th>Test RMSE with basic predictors</th>
<th>Test RMSE with basic p. and with SVD with biases</th>
<th>Cumulative test RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>six basic predictors</td>
<td>.9826</td>
<td>.9039</td>
<td>.9826</td>
</tr>
<tr>
<td>regularized SVD</td>
<td>.9094</td>
<td>.9018</td>
<td>.9094</td>
</tr>
<tr>
<td>reg. SVD with biases</td>
<td>.9039</td>
<td>.9039</td>
<td>.9018</td>
</tr>
<tr>
<td>K-means</td>
<td>.9410</td>
<td>.9029</td>
<td>.9010</td>
</tr>
<tr>
<td>postproc. SVD with 1-NN</td>
<td>.9525</td>
<td>.9013</td>
<td>.8988</td>
</tr>
<tr>
<td>postproc. SVD with KRR</td>
<td>.<strong>9006</strong></td>
<td><strong>.8959</strong></td>
<td>.8933</td>
</tr>
<tr>
<td>weighted linear models</td>
<td>.9506</td>
<td>.8995</td>
<td>.8902</td>
</tr>
<tr>
<td>small SVD-based 1</td>
<td>.9312</td>
<td>.8986</td>
<td>.8887</td>
</tr>
<tr>
<td>small SVD-based 2</td>
<td>.9590</td>
<td>.9032</td>
<td>.8879</td>
</tr>
</tbody>
</table>

**Table:** Linear regression results - RMSE on the test set
Experimental results

- the method with the best results: postprocessing SVD with kernel ridge regression
- combining all methods from the table plus 2 two-way interactions gives 6.34% improvement on the validation set (qualifying.txt)
- the 7.04% solution is a result of combining 56 predictors and 63 two-way interactions
- running times varied from 45min to 20h on a PC with 2GHz processor and 1.2GB RAM
Conclusions

• combining many methods with linear regression on the test set is a very effective approach

• good results of postprocessing SVD with kernel ridge regression suggest possibility of developing better methods than SVD for collaborative filtering